

REMARKS

Claims 16 and 28 have been amended. Claims 16-53 remain pending in the application. Reconsideration is respectfully requested in light of the following remarks.

Section 112, Second Paragraph, Rejection:

The Examiner rejected claims 28-39 and 47-53 under 35 U.S.C. § 112, second paragraph, as indefinite. Applicants submit that amended independent claim 28 addresses the 112 rejection. Removal of this rejection is respectfully requested.

Section 103(a) Rejection:

The Examiner rejected claims 16-18, 20, 21, 23, 24, 26-30, 32, 33, 35, 36, 38-41, 43, 44, 46, 47, 49, 50, 52 and 53 under 35 U.S.C. § 103(a) as being unpatentable over Simser (U.S. Patent 6,314,429) in view of Krapf et al. (U.S. Patent 6,901,588) (hereinafter “Krapf”), and claims 19, 22, 25, 31, 34, 37, 42, 45, 48 and 51 as being unpatentable over Simser and Krapf further in view of Goldsmith (U.S. Patent 5,491,800). For at least the reasons presented below, Applicants respectfully traverse these rejections.

Regarding claim 16, Simser in view of Krapf fails to teach or suggest that the second mediation module is configured to send a request for a function reference to the first mediation module, wherein the request comprises one or more of: a package name, a class name, and a method name of a method to be invoked, and wherein in response to receiving the request, the first mediation module is configured to return the function reference to the second mediation module. Neither Simser nor Krapf, whether considered singly or in combination mention anything about one mediation module requesting a function reference from another mediation module. In contrast, Simser teaches the use of the Java Native Interface (JNI) and DLLs implementing API's for translating data types between C++ and JAVA. Nowhere does Simser mention a mediation module requesting

or receiving a function reference. Similarly, Krapf teaches the use of proxy interfaces to make function calls between C++ and JAVA. Krapf's proxy interfaces do not request or receive function references. Thus, the combination of Simser and Krapf fails to teach or suggest a mediation module configured to request and receive a function reference.

Furthermore, Simser in view of Krapf nowhere disclose communicating the function call to the first mediation module using the function reference. The Examiner relies on Krapf to disclose a first and second mediation module through Krapf's proxy interfaces. As noted above, the Java and C++ interfaces disclosed by Krapf do not use function references. Neither Simser nor Krapf, taken singly or in combination, teach or suggest *communicating the function call to a mediation module using the function reference*. Thus, the rejection of claim 16 is not supported by the prior art and removal thereof is respectfully requested. Similar remarks as those above regarding claim 16 also apply to claim 28.

Regarding claim 40, Simser fails to anticipate a platform independent language application running on a computer initiating an instance of a native language application, contrary to the Examiner's assertion. The Examiner refers to Simser's teachings regarding calling an API of a legacy application and cites column 3, lines 1-15 of Simser. However, the cited passage does not mention a platform independent language application initiating an instance of a native language application. Instead, the cited passage merely refers to a Java application calling functions in a legacy application. Nowhere does Simser teach that the Java application initiates an instance of the legacy application. Merely calling a function in a legacy application does not imply initiating an instance of the legacy application by the platform independent language application. To the contrary, the legacy application in Simser would have to already be initiated in order for the Java application in Simser to call one of its functions. Thus, Simser clearly fails to disclose a platform independent language application running on a computer initiating an instance of a native language, as recited in claim 40.

In further regard to claim 40, contrary to the Examiner's assertion in the Response to Arguments, Krapf also fails to disclose a platform independent language application running on a computer initiating an instance of a native language application. The Examiner cites column 42, lines 23-33 of Krapf:

During execution of the C++ application, an instance of C++ class 517 is constructed using constructor 518. Execution of constructor 518 causes the execution of C++ superclass 510's constructor 512, which, in turn, causes the creation of an instance of Java class 499 as illustrated in C++ code fragment 513.

The Examiner has erroneously characterized the construction of a class as initiating an instance of a native language application. Anyone skilled in the art of computer programming understands that class construction is not initiating an instance of a native language application. Nowhere does Krapf disclose this limitation of claim 40. Thus, Krapf clearly fails to disclose a platform independent language application running on a computer initiating an instance of a native language as recited by claim 40.

Thus, neither Simser nor Krapf, taken singly or in combination, disclose a platform independent language application running on a computer initiating an instance of a native language, as recited in claim 40.

Thus, for at least the reasons above, the rejection of claim 40 is not supported by the prior art and removal thereof is respectfully requested.

Applicants also assert that numerous ones of the dependent claims recite further distinctions over the cited art. However, since the rejection has been shown to be unsupported for the independent claims, a further discussion of the dependent claims is not necessary at this time.

CONCLUSION

Applicants submit the application is in condition for allowance, and notice to that effect is respectfully requested.

If any fees are due, the Commissioner is authorized to charge said fees to Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C. Deposit Account No. 501505/5681-78801/RCK.

Also enclosed herewith are the following items:

☒ Return Receipt Postcard

Respectfully submitted,



Robert C. Kowert

Reg. No. 39,255

ATTORNEY FOR APPLICANT(S)

Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C.
P.O. Box 398
Austin, TX 78767-0398
Phone: (512) 853-8850

Date: November 10, 2005